

# Loss Function: Mean Square Error

Ki Hyun Kim

[nlp.with.deep.learning@gmail.com](mailto:nlp.with.deep.learning@gmail.com)

# Again, Our Objective is

- 데이터를 넣었을 때 출력을 반환하는 가상의 함수를 **모사**하는 것
- Linear Layer 함수를 통해 원하는 함수를 모사해보자
  - Linear Layer 함수가 얼마나 원하는 만큼 동작하는지 측정하고 싶다.
  - 잘 동작하는지, 점수로 나타내고 싶다.

# Loss

- Loss(손실값): 원하는 출력값(target,  $y$ )과 실제 출력값(output,  $\hat{y}$ )의 차이의 합

$$\begin{aligned}\text{Loss} &= \sum_{i=1}^N \|y_i - \hat{y}_i\| \\ &= \sum_{i=1}^N \|y_i - f(x_i)\|\end{aligned}$$

- 그러므로 우리는 Loss가 작을 수록 가상의 함수를 잘 모사하고 있다고 할 수 있음
- Loss가 작은 Linear Layer를 선택하면 됨

# Loss Function

- Linear Layer의 파라미터를 바꿀때마다 Loss를 계산
- Loss Function
  - 입력: Linear Layer의 파라미터
  - 출력: Loss

$$\mathcal{L}(\theta) = \sum_{i=1}^N \|y_i - f_{\theta}(x_i)\|,$$

where  $\theta = \{W, b\}$ .

# Euclidean Distance

- 두 점 사이의 거리

$$\begin{aligned}\|y - \hat{y}\|_2 &= \sqrt{(y_1 - \hat{y}_1)^2 + \cdots + (y_n - \hat{y}_n)^2}, \\ &= \sqrt{\sum_{i=1}^n (y_i - \hat{y}_i)^2}\end{aligned}$$

where  $y \in \mathbb{R}^n$  and  $\hat{y} \in \mathbb{R}^n$ .

# RMSE (Root Mean Square Error)

- Euclidean distance와 비슷한 개념

$$\text{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

# MSE (Mean Square Error)

- Root와 상수를 뺐지만, 크기 차이로 인한 순서 결과는 바뀌지 않음

$$\begin{aligned}\text{MSE}(y, \hat{y}) &= \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \\ &= \frac{1}{n} (\|y - \hat{y}\|_2)^2 \\ &= \frac{1}{n} \|y - \hat{y}\|_2^2 \\ &\propto \|y - \hat{y}\|_2^2\end{aligned}$$

# Conclusion

- 우리는 목표로 하는 함수를 모사하기 위해,
  - 학습용 입력 데이터들을 Linear Layer에 넣어 출력 값들을 구하고,
  - 출력 값( $\hat{y}$ )들과 목표 값( $y$ )들의 차이의 합(Loss)을 최소화 해야한다.
- 결국, Linear Layer 파라미터( $\theta$ )를 바꾸면서 Loss를 최소화 해야한다.